ISSN 1009 - 3095 *Journal of Zhejiang University(SCIENCE)* V.1, No.3, P.264 - 268, July - Sep., 2000
http://www.chinainfo.gov.cn/periodical; http://www.zju.edu.cn;
http://lib.zju.edu.cn/English; http://www.zjupress.com

FAST ALGORITHM FOR DCT DOMAIN IMAGE UP-CONVERSION

LI Hong-dong(李宏东), XIE Rong(解蓉), LIU Ji-lin(刘济林), GU Wei-kang(顾伟康)

(Dept. of Information and Electronics, Zhejiang University, Hangzhou 310027, China)

E-mail: {lihd, xier, liujl} @isee.zju.edu.cn; guwk@sun.zju.edu.cn

Received May 22, 1999; revision accepted Jan. 6, 2000

Abstract: A great number of data associated with visual information requires usage of the popular DCT (Discrete Cosine Transform) compression. This also suggests the investigation of methods for image processing directly in the DCT domain. This paper presents a fast DCT domain image up-conversion algorithm, and a DCT domain linear filter, which only involves matrix manipulations. By using the separate property of certain spatial filters, we accomplish the DCT domain up-conversion via only matrix products. Because 1) most of the matrices can be calculated beforehand, and 2) common DCT blocks are very sparse; and 3) by avoiding the conventional decode and re-encode scheme, the computational complexities are greatly reduced. Experiments on real images show good results.

Key words: compressed domain process, image up-conversion, JPEG, standard conversion, image interpolation, DCT domain filtering
Document code: A **CLC number:**

INTRODUCTION

There are constantly increasing demands in recent years for multimedia applications, various image/video-processing methods including image enhancement and restoration, video composition, color compensation, and sample structure conversion (down-conversion and up-conversion), etc. Many methods require real-time manipulation in order to implement special effects without delay.

Due to the huge volume of stored data and their transmission requirements, image/video compression techniques are becoming more and more necessary. As multimedia is now predominantly stored in compressed form, the manipulation of compressed data must be taken into consideration. The greatly reduced image and video data after they are compressed suggests the potential to reach real-time realization. Since the Discrete Cosine Transform (DCT) is central to many popular compression methods, in this work we focus our research on DCT.

The traditional image processing algorithms for calculation of compressed images/videos usually first decompress the data, then manipulate them in the decompressed domain, and finally, compress again if necessary introduce a great amount of extra computation and considerable delays, which will not meet the real-time requirement. For this reason there has been great effort in recent years to develop fast algorithms for manipulating the compressed data directly in the compressed domain (Assuncao et al., 1998; Merhav et al., 1996; Assuncao et al., 1997; Chang et al., 1995). And thereby avoid the need for decompression or at least its computational bottleneck, the Inverse DCT (IDCT) which requires 38.7% of the execution time on a typical workstation (Merhav, et al., 1997).

This paper focuses on the problem of image up-conversion, which is important for standard conversion. Examples are PAL/NTSC to HDTV, and MPEG-2 MP@ML (HL) to 4:2:2P@ML (HL) etc. Most traditional up-conversion algorithms are mainly for use in spatial domain via pixel interpolation techniques, with the following typical methods having been introduced: the Nearest Neighbor (NN), Bilinear (BL), and Cubic Spline(CS), etc. Non-intuitive, up-conversion acting directly on compressed data will have the advantage of less computational burden, as well as the flexibility to accommodate dynamic and heterogeneous resources.

There were some reports of work on DCT domain processing, in particular, on sample structure conversion, but most of them were on down-conversion (Neri et al., 1994; Merhav et al., 1997), as it is easier to realize. To our knowledge, there are few reports in literature of up-conversion algorithms, which have the same importance as the down-conversion algorithms. In this paper, we present a simple DCT domain up-conversion algorithm that takes advantage of some nice properties of the DCT transform and the spatial domain filters. We used some pre-calculated matrices to design a simple but effective DCT domain up-conversion algorithm, as well as a DCT domain linear filter.

The paper is organized as follows. Section 2 gives a brief description of DCT and its properties. The proposed DCT domain up-conversion algorithm and DCT domain filtering is introduced in Section 3 and Section 4 respectively. Section 5 gives some experimental results, and the last section, Section 6, is the conclusion.

DEFINITION AND PROPERTIES OF DCT

DCT has the best energy compaction among all transforms and is close to the optimum K-L transform. It is used in many image/video compression standards including JPEG, MPEG-1,2, etc.

The 8 × 8 2-D DCT transforms an image block $\{x(n,m),\ m,n = 0,1,2,\cdots,7\}$ into a matrix of coefficients $\{X(k,l),\ k,l = 0,1,2,\cdots,7\}$ and the Inverse DCT are described as:

$$X(k,l) = \frac{c(k)}{2}\frac{c(l)}{2}\sum_{n=0}^{7}\sum_{m=0}^{7}x(n,m)\cdot \cos\left(\frac{2n+1}{16}k\pi\right)\cos\left(\frac{2m+1}{16}l\pi\right) \tag{1}$$

$$x(n,m) = \sum_{k=0}^{7}\sum_{l=0}^{7}\frac{c(k)}{2}\frac{c(l)}{2}X(k,l)\cdot \cos\left(\frac{2n+1}{16}k\pi\right)\cos\left(\frac{2m+1}{16}l\pi\right) \tag{2}$$

where $c(0) = 1/\sqrt{2}$, and $c(k) = 1$, for $k > 0$.

Both of the equations can be described in the following matrix form. Let $\boldsymbol{x} = \{x(n,m)\}$, and $\boldsymbol{X} = \{X(k,l)\}$ (where bold face denotes a matrix, and capital form stands for DCT component.). We define the 8 × 8 DCT transform $\boldsymbol{T}[\cdot]$, and the corresponding transform matrix $\boldsymbol{S}$, where

$$S(k,n) = \frac{c(k)}{2}\cos\left(\frac{2n+1}{16}k\pi\right) \tag{3}$$

Then $\quad \boldsymbol{X} = \boldsymbol{T}[\boldsymbol{x}],\ \boldsymbol{x} = \boldsymbol{T}^{-1}[\boldsymbol{X}] \quad (4)$

And $\quad \boldsymbol{X} = \boldsymbol{S}\cdot\boldsymbol{x}\cdot\boldsymbol{S}^T, \quad (5)$

Where the superscript T of matrix denotes transposition. Similarly, for inverse transform,

$$\boldsymbol{x} = \boldsymbol{S}^{-1}\cdot\boldsymbol{X}\cdot\boldsymbol{S}^{-T} = \boldsymbol{S}^T\cdot\boldsymbol{X}\cdot\boldsymbol{S} \tag{6}$$

where the second equality follows from the property of unitarity of $\boldsymbol{S}$, i.e., $\boldsymbol{S}^{-1} = \boldsymbol{S}^T$.

Also we can conclude from Eq.5 and Eq.6 that the DCT and IDCT are both linear transforms. So they have such properties of distributive as,

- Distributive to matrix addition
- Distributive to matrix multiplication

Written in formula form, they are:

$$\boldsymbol{T}[\boldsymbol{A}\cdot\boldsymbol{B}] = \boldsymbol{T}[\boldsymbol{A}]\cdot\boldsymbol{T}[\boldsymbol{B}]$$
$$\boldsymbol{T}[\boldsymbol{A}+\boldsymbol{B}] = \boldsymbol{T}[\boldsymbol{A}]+\boldsymbol{T}[\boldsymbol{B}] \tag{7}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are arbitrary matrices.

These properties are useful for deriving many linear DCT domain manipulations since they can be modeled in some forms of matrix multiplication. Because of the energy compactness, a quantized DCT matrix will become a sparse matrix, which will greatly reduce the involved matrix computation.

DCT DOMAIN UP-CONVERSION

Traditional up-conversion has been primary performed in spatial domain via pixel interpolation techniques. Many methods have been introduced. The most popular are the Nearest Neighborhood (NN), the Bilinear (BL), and the Cubic Spline (CS), etc., as well as some other brute force but complicated algorithms, such as POCS, MRF, nonlinear diffusion, etc. (Chang et al., 1995).

Most of the methods are based on the expectation that the image to be interpolated is very smooth. This results in a Low Pass Filter (LPF) scheme for interpolation (Chiptrasert et al., 1990). Besides, this LPF can be written in matrix form that provides the possibility of simplicity of the corresponding transform domain manipulation. We will explain it below in detail. Fig.1 depicts this problem. What we want to find is the mathematical equivalent counterpart of the spatial processing.

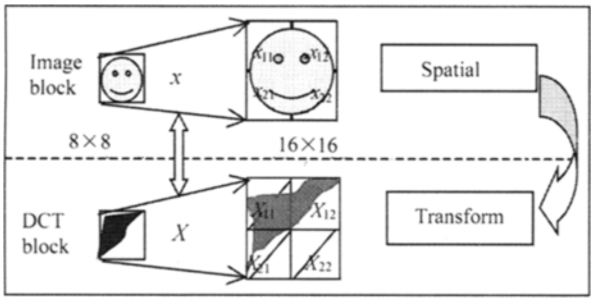

Fig.1 Spatial and DCT domain up conversion

Fortunately, the spatial LPF for interpolation is separable, which means that we can first enlarge the original image block vertically, then followed by an enlargement in horizontal direction. Let x_{11}, x_{12}, x_{21}, x_{22} (shown in Fig. 1) denote the resultant 4 adjacent blocks of the interpolation of image block x, and the corresponding DCT domain counterpart X_{11}, X_{12}, X_{21}, X_{22}. According to the separable property, we then obtain the following spatial domain up-conversion formulas,

$$
\begin{aligned}
x_{11} &= A_1 \cdot x \cdot A_1^T \\
x_{12} &= A_1 \cdot x \cdot A_2^T \\
x_{21} &= A_2 \cdot x \cdot A_1^T \\
x_{22} &= A_2 \cdot x \cdot A_2^T
\end{aligned} \tag{8}
$$

where A_1 and A_2 are primary matrices used for structure interpolation. The left matrix means vertical interpolating, and the right one means horizontal interpolating. For instance, if the up-conversion structure is shown as Fig.2, where each of the new pixel values is a linear combination of its neighbors', we can readily write down the A_1 and A_2 as:

$$
A_1 = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.75 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.25 & 0.75 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.75 & 0.25 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.25 & 0.75 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.75 & 0.25 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.25 & 0.75 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.75 & 0.25 & 0 & 0 & 0
\end{bmatrix},
$$

$$
\mathrm{A}_2 = \mathrm{U} * \mathrm{A}_1 * \mathrm{U} \tag{9}
$$

$$
\text{where} \quad U = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

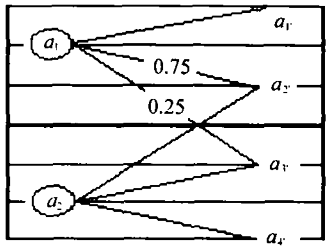

Fig.2 Up conversion structure

According to the distributive properties in Eq.7, we thus get the DCT domain up-conversion formulas,

$$
\begin{aligned}
X_{11} &= T[x_{11}] = T[A_1] \cdot X \cdot T[A_1^T] \\
X_{12} &= T[x_{12}] = T[A_1] \cdot X \cdot T[A_2^T] \\
X_{21} &= T[x_{21}] = T[A_2] \cdot X \cdot T[A_1^T] \\
X_{22} &= T[x_{22}] = T[A_2] \cdot X \cdot T[A_2^T]
\end{aligned} \tag{10}
$$

Because we can calculate the 8×8 matrices $T[A_1]$, $T[A_1^T]$, $T[A_2]$, $T[A_2^T]$ in advance, and store them in memory, the only involved calculation during the DCT domain up-conversion procedure is matrix multiplication. Moreover, some techniques for sparse matrix manipulation will reduce further the computational complexity, since many DCT coefficients are zero.

DCT DOMAIN LINEAR FILTERING

Improvement of the image quality after up-conversion often requires spatial domain filtering on the obtained image blocks. This filtering can also be achieved in DCT domain, due to the linear and separable properties of certain filters. Because the filtering process in spatial domain actually is a convolution operation, several input blocks will contribute to the output block. Thus in matrix form, a linear filter is described by a summation of matrix multiplication,

$$B = \sum_i L_i A_i R_i \tag{11}$$

where $\mathbf{A}_i$ is the ith input block, and the summation range depends on the filter length.

Many widely used image filter kernels such as box averaging and Gaussian smoothing are separable. Thus, it can be treated in a way similar to up-conversion. So, a linear filtering of images in the horizontal direction can be achieved by a summation of right matrix products, and the left corresponds to the vertical filtering. On the basis of our experiments, we chose the following smoothing filter,

$$K = \frac{1}{100}\begin{bmatrix} 1 & 8 & 1 \\ 8 & 64 & 8 \\ 1 & 8 & 1 \end{bmatrix}, K_L = [1 \quad 8 \quad 1]^{\mathrm{T}},$$
$$K_R = [1 \quad 8 \quad 1], \frac{1}{100}K = K_L \otimes K_R, \tag{12}$$

where '$\otimes$' denotes the Kronecker product.

In this paper, we ignore the outside neighbor blocks, for their contribution to the summation is relatively small, which only causes minor blocky effects. Thus the filtering can be achieved via only one left matrix and one right matrix. By Eq.12, we give the right matrix $\boldsymbol{R}$ for example.

$$R = \frac{1}{10}\begin{bmatrix} 8 & 1 & & & & & 0 \\ 1 & 8 & 1 & & & & \\ & 1 & 8 & 1 & & & \\ & & 1 & 8 & 1 & & \\ & & & 1 & 8 & 1 & \\ & & & & 1 & 8 & 1 \\ 0 & & & & & 1 & 8 & 1 \end{bmatrix}$$

Since the DCT algorithm is distributive to matrix multiplication, we can calculate the corresponding DCT filtering in a similar way like Eq.10. In addition, the computation of both the filtering and the above up-conversion in DCT domain can be absorbed into one procedure, which results in the final formula as:

$$\begin{aligned} X_{ij} &= T[L \cdot A_i \cdot x_{ij} \cdot A_j^T \cdot R] \\ &= T[L \cdot A_i] \cdot X \cdot T[A_j^T \cdot R] \end{aligned} \tag{13}$$

where $i, j = 1, 2$. Both $T[L \cdot A_i]$ and $T[A_j^T \cdot R]$ can be pre-calculated.

After the DCT domain up-conversion and smoothing, the resultant DCT coefficients should be quantized again for the purpose of data reduction, and can be decoded by other applications.

EXPERIMENTAL RESULTS

This section gives some experiments on real images to test the proposed up-conversion algorithm. The inputs were some size 256 × 256 standard real images called as original images. We first sub-sampled them in spatial domain into 128 × 128 size and then compressed them by a baseline JPEG algorithm. Our algorithm depicted in Fig.3 is thus acting on these 128 × 128 JPEG images, as shown in Fig.3.

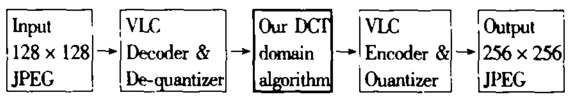

Fig.3 Flow chart of the DCT domain up-conversion

Fig.4 shows the JPEG 128 × 128 images, which are 'Lena', 'girl' and 'cameraman'. Fig.5 shows the resultant up-sampled images. For the sake of comparison, we also give the results obtained by the spatial domain methods such as NN and BL. Table1 shows the PSNR results, for comparison with those of size 256 × 256 original images.

Fig.4 The JPEG images of 128 × 128

Table 1 Comparison of PSNR

	PSNR (dB)		
	NN	BL	Our algorithm
Lena128.jpg	18.4040	19.7273	19.3472
Girl128.jpg	24.4979	26.0161	25.9914
Cameraman.jpg	23.2015	26.6701	25.6463

From this table, we can conclude that the performance of the proposed DCT domain up-conversion algorithm is fairly good, when compared with spatial domain interpolating. But our algorithm works much faster than the spatial methods when used on compressed images.

Fig. 5 **Ther resultant** 256 × 256 **images by our algorithm**

CONCLUSION

The fast algorithm for DCT domain image up-conversion proposed in this paper takes advantage of the nice properties of DCT transform and spatial filters. Due to the much lower data rate in the compressed domain, this type of manipulation provides great potential for reducing the overall computational complexity. It realizes this by dispensing with not only very time-consuming decode and re-encode schemes, but also some useless computation because of the sparseness of DCT blocks. Good use of the special strategies for sparse matrix manipulations further lightens the computation burden. The reduced processing delay reduces the overall system latency as well as the memory requirements. We have tested the method on various types of real images. The very satisfactory results obtained confirmed the good performance of our algorithm.

References

Assuncao, P.A.A. Ghanbari, M., 1998. A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams. *IEEE Transactions on Circuits and Systems for Video Technology*, **8**:953 – 967.

Assuncao, P.A.A. Ghanbari, M., 1997. Fast computation of MC – DCT for video transcoding. *Electronics Letters*, **33**:284 – 286.

Chang, S. F. Messerschmitt, D. G., 1995. Manipulation and composing of MC – DCT compressed video. *IEEE J-Sel. Area on Commu*, **13**(1):1 – 11.

Chiptrasert, B. Rao, K.R., 1990. Discrete cosine transform filtering. *Sig. Proc*.**19**(3):233 – 245.

Merhav, N. Bhaskaran, V., 1996. A fast algorithm for DCT-domain inverse motion compensation, IEEE International Conference on Acoustics. *Speech, and Signal Processing*, **4**(4):2307 – 2310.

Merhav, N. Bhaskaran, V., 1997. Fast algorithms for DCT domain image down-conversion and for inverse motion compensation. *IEEE T- CSVT*, **7**(3):468 – 476.

Neri, A. Russo, G. Talone, P., 1994. Inter-block filtering and down conversion in DCT domain. *Sig. Proc.: Image Communication*, **6**:303 – 317.